





in React

codewithsloba.com

Swipe







Custom hooks are reusable functions that allow you to extract and reuse stateful logic from components.

They are a way to abstract away complex logic into a separate function that can be shared across multiple components.

Swipe for example!





Lets say we want to display username and age, how can we do that using a custom hook?

We can create a custom hook for that and call it useUserData

```
useUserData.js
initializing with
useState

import { useState } from 'react';

const useUserData = () => {
  const [username, setUsername] = useState('');
  const [age, setAge] = useState(0);

  return { username, age, setUsername, setAge };
};

export default useUserData;
```

We also provide the respective setter functions, setUsername and setAge, to update the state values.





```
import React from 'react';
import useUserData from './useUserData';

const UserForm = () => {
   const { username, age, setUsername, setAge } =
   useUserData();

const handleUsernameChange = (e) => {
    setUsername(e.target.value);
   };

const handleAgeChange = (e) => {
   setAge(parseInt(e.target.value));
  };
}
```

The handleUsernameChange and handleAgeChange functions are responsible for updating the state based on user input





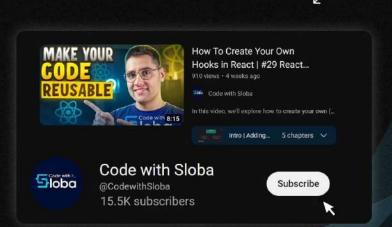
```
return (
      <label>Username:</label>
      <input type="text" value={username}</pre>
onChange={handleUsernameChange} />
      <label>Age:</label>
      <input type="number" value={age}</pre>
onChange={handleAgeChange} />
      {username && age && (
          {username} is {age} years old
        )}
    </div>
  );
};
export default UserForm;
```

Username: Bob
Age: 56
Bob is 56 years old

output 1



To better understand this topic check out this video on my **YouTube** channel.







codewithsloba.com

Get a weekly digest of my tips and tutorials by subscribing now.

